IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application No.: 10/673,088 Filed: September 26, 2003 Inventor(s): Michael A. Wasserman, et al. Title: LARGE-KERNEL CONVOLUTION USING MULTIPLE INDUSTRY-STANDARD GRAPHICS ACCELERATORS	§ § § § § § § § § § § § § §	Examiner: Group/Art Unit: Atty. Dkt. No:	Nguyen, Hau H. 2628 5681-59600

AMENDED APPEAL BRIEF SUBMITTED IN RESPONSE TO NOTICE OF NON-COMPLIANT APPEAL BRIEF

Dear Sir/Madam:

This Amended Appeal Brief is submitted in response to the Notice of Non-Compliant Appeal Brief mailed on August 1, 2007. The Notice indicates that Appellant omitted a statement regarding the status of Claims 15-17 and 20 in the Appeal Brief submitted on April 9, 2007. This Amended Appeal Brief corrects the omission. Appellant respectfully requests that this appeal be considered by the Board of Patent Appeals and Interferences.

I. REAL PARTY IN INTEREST

The subject application is owned by SUN MICROSYSTEMS, INC., a corporation organized and existing under and by virtue of the laws of the State of Delaware, and having its principal place of business at 901 San Antonio Road, Palo Alto, CA 94303, as evidenced by the assignment recorded at Reel 015192, Frame 0185.

II. RELATED APPEALS AND INTERFERENCES

No other appeals, interferences or judicial proceedings are known which would be related to, directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

Claims 1-25 are pending, and claims 1-14, 18-19, and 21-25 are rejected. Claims 15-17 and 20 are objected to as being dependent upon a rejected base claim, but are indicated as being allowable if rewritten to include all of the limitations of the base claim and any intervening claims. The rejection of claims 1-14, 18-19, and 21-25 is being appealed. A copy of claims 1-25 is included in the Claims Appendix hereto.

IV. STATUS OF AMENDMENTS

No amendments to the claims have been submitted subsequent to the final rejection.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Independent claim 1 is directed to a system for distributed convolution of stacked digital video data. The system comprises a plurality of video data convolve units connected in a chain. Each video data convolve unit in the chain is operable to: 1) receive video pixel data from a video output of a dedicated rendering unit; 2) calculate partial convolution sums for a set of the video pixels that are located within a convolution

kernel; 3) receive accumulated partial convolution sums from a prior video data convolve unit in the chain, unless the video data convolve unit is the first video data convolve unit in the chain; 4) add the calculated partial convolution sums to the previously accumulated partial convolution sums; and 5) output new accumulated partial convolution sums to the next video data convolve unit in the chain, unless the video data convolve unit is the last video data convolve unit in the chain (*as disclosed at least in Fig. 17 and at page 29, line 16 through page 31, line 22, of the specification*).

Independent claim 7 is directed to a system for convolution of tiled digital video data. The system comprises a plurality of video data convolve units connected in a chain. Each video data convolve unit in the chain is operable to: 1) receive video pixels from a video output of a graphics rendering unit; and 2) convolve a set of the video pixels that are located within a convolution kernel to determine values for a convolved video pixel corresponding to the convolution kernel, where the value for each pixel parameter equals a sum of weighted video pixel values for the parameter divided by a sum of weights, and where the weights are determined for locations of each video pixel in the set of video pixels (*as disclosed at least in Fig. 16 and at page 27, line 19 through page 29, line 12, of the specification*).

Independent claim 10 is directed to a system for distributed convolution of stacked and tiled digital video data. The system comprises a plurality of video data convolve units connected in a chain, where the video data convolve units are separated into a plurality of groups, where at least one of the groups has a plurality of video data convolve units, and where each group is operable to determine values for convolved video pixels located in a portion of screen space assigned to the group. Each video data convolve unit within a group is operable to: 1) receive video pixels from a video output of a dedicated rendering unit; 2) calculate partial convolution sums for a set of the video pixels that are located within a convolution kernel corresponding to a convolved pixel location; 3) receive accumulated partial convolution sums from a prior video data convolve unit in the chain, unless the video data convolve unit is the first video data convolve unit within the group; 4) add the calculated partial convolution sums to the accumulated partial convolution sums; and 5) output new accumulated partial convolution sums to the next video data convolve unit in the chain, unless the video data

convolve unit is the last video data convolve unit within the group (*as disclosed at least in Fig. 17 and at page 29, line 16 through page 31, line 22, of the specification*).

Independent claim 14 is directed to a method for distributed convolution of stacked digital video data in a plurality of video data convolve units connected in a chain. The method comprises (for each video data convolve unit): 1) receiving video pixel data from a video output of a dedicated rendering unit; 2) storing the video pixel data in a video line buffer; 3) performing a partial convolution as part of a distributed process to determine values for a convolved pixel by calculating partial convolution sums for the pixels in the line buffer that are located within a convolution kernel corresponding to the location of a convolved pixel; 4) adding the partial convolution sums to any corresponding accumulated partial convolution sums received from a prior video data convolve unit in the chain to form new accumulated partial convolution sums, unless the video data convolve unit is the first video data convolve unit in the chain; and 5) sending the new accumulated partial convolution sums to the next video data convolve unit in the chain, unless the video data convolve unit is the last video data convolve unit in the chain (*as disclosed at least in Fig. 18 and at page 31, line 26 through page 33, line 15, of the specification*).

Independent claim 22 is directed to a method for convolution of tiled digital video data in a plurality of video data convolve units connected in a chain. The method comprises (for each video data convolve unit): 1) receiving video pixel data from a video output of a dedicated rendering unit for a specified portion of screen space; 2) storing the video pixel data in a video line buffer; 3) determining values for a convolved pixel by convolving the video pixels in the line buffer that are located within a convolution kernel corresponding to the location of a convolved pixel; 4) storing the convolved pixel in a convolved pixel buffer; 5) combining the convolved pixels in the pixel buffer with convolved pixels received from a prior video data convolve unit so that the combined convolved pixels are ordered by their locations in a line of screen space, unless the video data convolve unit is the first video data convolve unit in the chain; and 6) sending the combined convolved pixels to the next video data convolve unit in the chain, unless the video data convolve unit is the last video data convolve unit in the chain (*as disclosed at least in Fig. 18 and at page 33, line 16 through page 34, line 5, of the specification*).

VI. <u>GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL</u>

Claims 1-14, 18-19, and 21-25 are rejected under 35 U.S.C. §103(a) as being unpatentable over Rousseau et al. (USPN 5524075) in view of Deering et al. (USPN 6417861).

VII. <u>ARGUMENT</u>

<u>Ground of Rejection</u>:

Claims 1-14, 18-19, and 21-25 are finally rejected under 35 U.S.C. §103(a) as being unpatentable over Rousseau et al. (USPN 5524075) (hereinafter "Rousseau") in view of Deering et al. (USPN 6417861) (hereinafter "Deering"). Appellant traverses this rejection for the following reasons. Different groups of claims are addressed under their respective subheadings.

<u>Claims 1-9</u>

Neither Rousseau nor Deering, either singly or in combination, teach or render obvious "…a plurality of video data convolve units connected in a chain, wherein a video data convolve unit is operable to: receive video pixel data from a video output of a dedicated rendering unit;…". The Examiner clearly states at page 3, lines 16-17 of the 11/14/06 Office Action that: "…**<u>Rousseau et al. do not teach receiving video pixel [data] from a video output of a dedicated rendering unit</u>**…". The Examiner further states at page 3, lines 17-18 of the 11/14/06 Office Action that "Deering teaches video data convolve units (170A-170D) receive **<u>pixel</u>** data from rendering units (Fig. 3, rendering unit[s] 150A-150D)". However, Deering does not teach that rendering units 150A-150D output video pixel data. Instead, Deering clearly identifies the output of the rendering units 150A-D as **<u>samples</u>** and specifically **<u>not pixels</u>** at col. 11, lines 18-22:

> "In the embodiment of graphics system 112 shown in the figure [Fig. 3], however, **<u>rendering units 150A-D calculate "samples" instead of actual pixel data</u>**. This allows rendering units 150A-D to "super-sample" or calculate more than one sample per pixel."

In addition, Deering teaches the sample output of the rendering units 150A-150D is stored in sample memories 160A-N, and sample-to-pixel calculation units 170A-D read the sample data from memory and convolve the sample data into pixel data (see Figure 3 of Deering). Therefore, Deering does not teach "receive video pixel data from a **<u>video output</u>** of a dedicated rendering unit". In fact, "video output" occurs only from the Digital to Analog Converters (DAC 178A-B) also shown in Fig. 3 of Deering.

Furthermore, neither Rousseau nor Deering, either singly or in combination, teach or render obvious "receive video pixel data from a video output of a **<u>dedicated</u>** rendering unit". As noted above (and shown in Fig. 3 of Deering), the rendering units 150A-150D of Deering output sample data to sample memories 160A-N and are not dedicated to any specific sample-to-pixel calculation unit. In addition, the video convolve units as taught by Rousseau (Fig. 6C, 3416s) receive data from a common bus (Fig. 6C, A Bus) and therefore a specific video convolve unit 3416 does not have a dedicated rendering unit.

Therefore, neither Rousseau nor Deering, either singly or in combination, teach or render obvious "a video data convolve unit is operable to: receive video pixel data from a video output of a dedicated rendering unit". Consequently, Applicant submits that claims 1 and 7 and their dependent claims are non-obvious and patentably distinguished over Rousseau and Deering for at least the reasons given above.

<u>Claims 10-13</u>

Neither Rousseau nor Deering, either singly or in combination, teach or render obvious "…a plurality of video data convolve units connected in a chain, wherein the video data convolve units are separated into a plurality of groups, … each video data convolve unit within a group is operable to: receive video pixel data from a **video output**

of a dedicated rendering unit;…". The Examiner clearly states at page 3, lines 16-17 of the 11/14/06 Office Action that: "…**Rousseau et al. do not teach receiving video pixel [data] from a video output of a dedicated rendering unit**…". The Examiner further states at page 3, lines 17-18 of the 11/14/06 Office Action that "Deering teaches video data convolve units (170A-170D) receive **pixel** data from rendering units (Fig. 3, rendering unit[s] 150A-150D)". However, Deering does not teach that rendering units 150A-150D output video pixel data. Instead, Deering clearly identifies the output of the rendering units 150A-D as **samples** and specifically **not pixels** at col. 11, lines 18-22:

> "In the embodiment of graphics system 112 shown in the figure [Fig. 3], however, **rendering units 150A-D calculate "samples" instead of actual pixel data**. This allows rendering units 150A-D to "super-sample" or calculate more than one sample per pixel."

In addition, Deering teaches the sample output of the rendering units 150A-150D is stored in sample memories 160A-N, and Sample-To-Pixel Calculation Units 170A-D read the sample data from memory and convolve the sample data into pixel data (see Figure 3 of Deering). Therefore, Deering does not teach "receive video pixel data from a **video output** of a dedicated rendering unit". In fact, "video output" occurs only from the Digital to Analog Converters (DAC 178A-B) also shown in Fig. 3 of Deering.

Furthermore, neither Rousseau nor Deering, either singly or in combination, teach or render obvious "receive video pixel data from a video output of a **dedicated** rendering unit". As noted above (and shown in Fig. 3 of Deering), the rendering units 150A-150D of Deering output sample data to sample memories 160A-N and are not dedicated to any specific sample-to-pixel calculation unit. In addition, the video convolve units as taught by Rousseau (Fig. 6C, 3416s) receive data from a common bus (Fig. 6C, A Bus) and therefore a specific video convolve unit 3416 does not have a dedicated rendering unit.

Therefore, neither Rousseau nor Deering, either singly or in combination, teach or render obvious "a plurality of video data convolve units connected in a chain, wherein the video data convolve units are separated into a plurality of groups, … each video data convolve unit within a group is operable to: receive video pixel data from a video output of a dedicated rendering unit". Consequently, Applicant submits that claim 10 and its

dependent claims are non-obvious and patentably distinguished over Rousseau and Deering for at least the reasons given above.

Claims 14-25

Neither Rousseau nor Deering, either singly or in combination, teach or render obvious "…a plurality of video data convolve units connected in a chain comprising (for each video data convolve unit): receiving video pixel data from a video output of a dedicated rendering unit; storing the video pixel data in a video line buffer…". The Examiner clearly states at page 3, lines 16-17 of the 11/14/06 Office Action that: "…**Rousseau et al. do not teach receiving video pixel [data] from a video output of a dedicated rendering unit**…". The Examiner further states at page 3, lines 17-18 of the 11/14/06 Office Action that "Deering teaches video data convolve units (170A-170D) receive **pixel** data from rendering units (Fig. 3, rendering unit[s] 150A-150D)". However, Deering does not teach that rendering units 150A-150D output video pixel data. Instead, Deering clearly identifies the output of the rendering units 150A-D as **samples** and specifically **not pixels** at col. 11, lines 18-22:

> "In the embodiment of graphics system 112 shown in the figure [Fig. 3], however, **rendering units 150A-D calculate "samples" instead of actual pixel data**. This allows rendering units 150A-D to "super-sample" or calculate more than one sample per pixel."

In addition, Deering teaches the sample output of the rendering units 150A-150D is stored in sample memories 160A-N, and Sample-To-Pixel Calculation Units 170A-D read the sample data from memory and convolve the sample data into pixel data (see Figure 3 of Deering). Therefore, Deering does not teach "receive video pixel data from a **video output** of a dedicated rendering unit". In fact, "video output" occurs only from the Digital to Analog Converters (DAC 178A-B) also shown in Fig. 3 of Deering.

Furthermore, neither Rousseau nor Deering, either singly or in combination, teach or render obvious "receive video pixel data from a video output of a **dedicated** rendering unit". As noted above (and shown in Fig. 3 of Deering), the rendering units 150A-150D

of Deering output sample data to sample memories 160A-N and are not dedicated to any specific sample-to-pixel calculation unit. In addition, the video convolve units as taught by Rousseau (Fig. 6C, 3416s) receive data from a common bus (Fig. 6C, A Bus) and therefore a specific video convolve unit 3416 does not have a dedicated rendering unit.

In addition, neither Deering nor Rousseau teach "storing the video pixel data in a video line buffer".

Therefore, neither Rousseau nor Deering, either singly or in combination, teach or render obvious "a plurality of video data convolve units connected in a chain comprising (for each video data convolve unit): receiving video pixel data from a video output of a dedicated rendering unit; storing the video pixel data in a video line buffer". Consequently, Applicant submits that claims 14 and 22 and their dependent claims are non-obvious and patentably distinguished over Rousseau and Deering for at least the reasons given above.

VIII. CONCLUSION

For the foregoing reasons, it is submitted that the Examiner's rejection of claims 1-14, 18-19, and 21-25 was erroneous, and reversal of his decision is respectfully requested.

The fee of $500.00 for filing this Appeal Brief is being paid concurrently via EFS-Web. If any extensions of time (under 37 C.F.R. § 1.136) are necessary to prevent the above-referenced application(s) from becoming abandoned, Applicant(s) hereby petition for such extensions. The Commissioner is hereby authorized to charge any fees which may be required or credit any overpayment to Meyertons, Hood, Kivlin, Kowert & Goetzel P.C., Deposit Account No. 50-1505/5681-59600/JCH.

Respectfully submitted,

/Mark K. Brightwell/
Mark K. Brightwell, Reg. #47446
AGENT FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert & Goetzel PC
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8800
Date: August 29, 2007 MKB

IX. CLAIMS APPENDIX

The claims on appeal are as follows.

1. (Original) A system for distributed convolution of stacked digital video data comprising:

   a plurality of video data convolve units connected in a chain, wherein a video data convolve unit is operable to:

   receive video pixel data from a video output of a dedicated rendering unit;

   calculate partial convolution sums for a set of the video pixels that are located within a convolution kernel;

   receive accumulated partial convolution sums from a prior video data convolve unit in the chain, unless the video data convolve unit is the first video data convolve unit in the chain;

   add the calculated partial convolution sums to the previously accumulated partial convolution sums; and

   output new accumulated partial convolution sums to the next video data convolve unit in the chain, unless the video data convolve unit is the last video data convolve unit in the chain.

2. (Original) The system of claim 1, further comprising one or more partial results buses, wherein each bus connects a video data convolve unit in the chain to a next video data convolve unit in the chain.

3. (Previously Presented) The system of claim 1, wherein the video data convolve unit is further operable to convert the format of the video pixel data to a digital data format utilized by the video data convolve unit.

4. (Original) The system of claim 1, wherein the partial convolution sums comprise (for each parameter value specified for each pixel) 1) a sum of weights determined for

locations of each video pixel in the set of video pixels and 2) a sum of weighted video pixel values for the set of video pixels.

5. (Original) The system of claim 1, wherein the video data convolve unit comprises a video line buffer utilized to store lines of video pixels received from the video output of the rendering unit.

6. (Original) The system of claim 5, wherein the video data convolve unit further comprises a convolution calculation unit that is operable to calculate partial convolution sums for the set of pixels, a partial results accumulator that is operable to add the partial convolution sums to corresponding partial results received and to output the new accumulated partial results, and a pixel value calculator that is operable in the last video data convolve unit in the chain to determine values for a convolved pixel from the final accumulated partial sums.

7. (Original) A system for convolution of tiled digital video data comprising:

   a plurality of video data convolve units connected in a chain, wherein each unit is operable to:

      receive video pixels from a video output of a graphics rendering unit; and

      convolve a set of the video pixels that are located within a convolution kernel to determine values for a convolved video pixel corresponding to the convolution kernel, wherein the value for each pixel parameter equals a sum of weighted video pixel values for the parameter divided by a sum of weights, wherein the weights are determined for locations of each video pixel in the set of video pixels.

8. (Original) The system of claim 7, further comprising one or more partial video buses, wherein each bus connects a video data convolve unit in the chain to a next video data convolve unit in the chain.

9. (Original) The system of claim 7, wherein the video data convolve unit comprises a video blend unit that is operable to receive convolved video pixels from a prior video data convolve unit and output a stream of convolved video pixels that is a combination of the received and generated video pixels ordered by screen location.

10. (Original) A system for distributed convolution of stacked and tiled digital video data comprising:

a plurality of video data convolve units connected in a chain, wherein the video data convolve units are separated into a plurality of groups, wherein at least one of the groups has a plurality of video data convolve units, and wherein each group is operable to determine values for convolved video pixels located in a portion of screen space assigned to the group; and

wherein each video data convolve unit within a group is operable to:

receive video pixels from a video output of a dedicated rendering unit;

calculate partial convolution sums for a set of the video pixels that are located within a convolution kernel corresponding to a convolved pixel location;

receive accumulated partial convolution sums from a prior video data convolve unit in the chain, unless the video data convolve unit is the first video data convolve unit within the group;

add the calculated partial convolution sums to the accumulated partial convolution sums; and

output new accumulated partial convolution sums to the next video data convolve unit in the chain, unless the video data convolve unit is the last video data convolve unit within the group.

11. (Original) The system of claim 10, further comprising one or more partial video buses, wherein each bus connects a last video data convolve unit in a group to a last video data convolve unit in the next group in the chain of video data convolve units.

12. (Original) The system of claim 10, further comprising one or more partial results buses, wherein each bus connects a video data convolve unit to a next video data convolve unit in a group.

13. (Original) The system of claim 10, wherein a video data convolve unit comprises a video blend unit, and wherein a last video data convolve unit in a group is operable to receive convolved video pixels from a prior group's last video data convolve unit and output a stream of convolved video pixels that is a combination of the received and generated convolved video pixels ordered by screen location.

14. (Original) A method for distributed convolution of stacked digital video data in a plurality of video data convolve units connected in a chain comprising (for each video data convolve unit):
receiving video pixel data from a video output of a dedicated rendering unit;
storing the video pixel data in a video line buffer;
performing a partial convolution as part of a distributed process to determine values for a convolved pixel by calculating partial convolution sums for the pixels in the line buffer that are located within a convolution kernel corresponding to the location of a convolved pixel;
adding the partial convolution sums to any corresponding accumulated partial convolution sums received from a prior video data convolve unit in the chain to form new accumulated partial convolution sums, unless the video data convolve unit is the first video data convolve unit in the chain; and
sending the new accumulated partial convolution sums to the next video data convolve unit in the chain, unless the video data convolve unit is the last video data convolve unit in the chain.

15. (Original) The method of claim 14, further comprising:
specifying a different jitter value or jitter pattern for each rendering unit;
sending vertex data for each geometric primitive to each rendering unit;
rendering pixel values for each jittered pixel location that lies within a geometric primitive; and

outputting the pixel values.

16. (Original) The method of claim 14, further comprising for the last video data convolve unit in the chain: determining parameter values for a convolved pixel from the final accumulated partial convolution sums, storing the convolved pixel values in a video output buffer, and outputting the convolved pixel data.

17. (Original) The method of claim 16, wherein determining parameter values for a convolved pixel comprises (for each parameter) dividing the final accumulated sum of weighted video pixel values for the parameter by a sum of weights, wherein the weights are determined for locations of each video pixel that is within the convolution kernel.

18. (Previously Presented) The method of claim 14, further comprising converting the video data output from the rendering unit to a digital data format utilized in the video data convolve unit if a format utilized for the video data output from the rendering unit differs from the digital data format utilized in the video data convolve unit.

19. (Original) The method of claim 14, wherein the partial convolution sums comprise 1) a sum of weights determined for locations of each video pixel in the set of video pixels and 2) a sum of weighted pixel values for the set of video pixels.

20. (Original) The method of claim 14, wherein the video pixel data from each rendering unit are determined for primitives that are geometrically expanded in both x and y dimensions by an integer factor of 2 or more; and wherein convolved pixel values are determined from the geometrically expanded pixel data and then assigned to convolved pixel locations determined by reducing the expanded locations by the same integer factor.

21. (Original) The method of claim 14, wherein each graphics rendering unit renders video pixels for primitives located anywhere in screen space.

22. (Original) A method for convolution of tiled digital video data in a plurality of video data convolve units connected in a chain comprising (for each video data convolve unit):

receiving video pixel data from a video output of a dedicated rendering unit for a specified portion of screen space;

storing the video pixel data in a video line buffer;

determining values for a convolved pixel by convolving the video pixels in the line buffer that are located within a convolution kernel corresponding to the location of a convolved pixel;

storing the convolved pixel in a convolved pixel buffer;

combining the convolved pixels in the pixel buffer with convolved pixels received from a prior video data convolve unit so that the combined convolved pixels are ordered by their locations in a line of screen space, unless the video data convolve unit is the first video data convolve unit in the chain; and

sending the combined convolved pixels to the next video data convolve unit in the chain, unless the video data convolve unit is the last video data convolve unit in the chain.

23. (Original) The method of claim 22, wherein a last video data convolve unit in the chain outputs the combined and ordered convolved video pixels to a display.

24. (Original) The method of claim 22, wherein each rendering unit renders video pixels for a different portion of screen space.

25. (Previously Presented) The method of claim 22, wherein frustum culling is utilized to sort the geometric primitives by screen portions and send each primitive to the corresponding rendering unit, and wherein those primitives that overlap a boundary between two screen portions may be sent to both corresponding rendering units or subdivided along the boundary.

X. EVIDENCE APPENDIX

No evidence submitted under 37 CFR §§ 1.130, 1.131 or 1.132 or otherwise entered by the Examiner is relied upon in this appeal.

XI. RELATED PROCEEDINGS APPENDIX

There are no related proceedings.